REMARKS

The present Amendment is in response to the Office Action having a mailing date of November 4, 2003. Claims 1-2 and 4-30 are pending in the present Application. Applicant has amended claims 1, 5, 7, 9, 13, 15, 17, 21, and 23. Applicant has also canceled claims 25-30 and added claims 31-39. Consequently, claims 1-2, 4-10, 12-18, 20-24, and 31-39 remain pending in the present Application.

Applicant has amended claims 1, 7, 9, 15, 17, and 23 to incorporate the limitations of claims 25, 26, 27, 28, 29, and 30 respectively. In particular, Applicant has amended claims 1, 7, 9, 15, 17, and 23 to recite that the subsequent increment is asynchronously formatted in response to occupation of a previous increment reaching a threshold that is less than one hundred percent occupation of the previous increment. Support for the amendment can be found in the specification, page 7, lines 4-8. Accordingly, Applicant respectfully submits that no new matter is added. Furthermore, claims 1, 7, 9, 15, 17, and 23 previously recited "asynchronously" formatting a subsequent increment. As used in the application asynchronous formatting occurs when formatting is not based upon the end of the previous increment being reached. Specification, page 7, lines 5-6. Thus, claims 1, 7, 9, 15, 17, and 23 previously indicated that formatting did not occur based upon the end, or one hundred percent occupation, of the previous increment being reached. Consequently, Applicant respectfully submits that the amendments to claims 1, 7, 9, 15, 17, and 23 do not narrow the scope of claims 1, 7, 9, 15, 17, and 23.

Applicant has also amended claims 5, 13, and 21 to more clearly recite that the formatting of the increment starts when data are first stored in a previous increment. Support for the amendment can be found in the specification, page 7, lines 5-6. Furthermore, because claims 5, 13,

and 21 previously recited formatting the increment when data are first stored in the previous increment, Applicant respectfully submits that no new matter is added.

Applicant has also added claims 31-39. Support for new claims 31-39 can be found in the specification, page 7, lines 4-8. Accordingly, Applicant respectfully submits that no new matter is added.

In the above-identified Office Action, the Examiner rejected claims 1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21, and 22 under 35 U.S.C. § 102 as being anticipated by U.S. Patent No. 6,172,955 (Hashimoto). The Examiner also rejected claims 3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23, and 24 under 35 U.S.C. § 103 as being obvious in light of Hashimoto in view of U.S. Patent No. 4,924,330 (Seamons).

In the above-identified Office Action, the Examiner rejected claims 1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21, and 22 under 35 U.S.C. § 102 as being anticipated by Hashimoto. In response to Applicant's arguments, the Examiner stated that "the claimed threshold is broad enough to read on the fact that once writing to a packet is complete, the method of Hashimoto formats a next packet (see column 9, lines 16-52)."

Applicant respectfully disagrees with the Examiner's rejection. Claim 1 recites a method for formatting space on a storage device for a database system. The method of claim 1 includes the step of "asynchronously formatting at least one subsequent increment in the storage device for the database file as a concurrent task of the database system . . ." Claim 1 further recites that the asynchronous formatting includes "formatting the at least one subsequent increment in response to occupation of a previous increment reaching a threshold of less than one hundred percent occupation of the previous increment." As indicated in the specification, page 7, lines 4-6, asynchronous formatting is formatting that is not based upon the end of the previous increment

being reached. To further clarify this, claim 1 recites that the asynchronous formatting is in response to a threshold of less than one hundred percent occupancy (i.e. before the end of the increment) being reached. Stated differently, in the method recited in claim 1, the threshold (of less than one hundred percent) occupation of an increment being reached causes formatting of the next increment to be commenced. Claims 9 and 17 recite analogous computer-readable medium and system claims, respectively.

Because the method, computer-readable medium and systems recited in claims 1, 9, and 17 asynchronously format increments in response to a threshold of less than one hundred percent occupation being reached, access delays due to formatting subsequent increments can be reduced or eliminated. Specification, page 6, lines 10-11; page 7, lines 10-11; and page 9, lines 1-2. Furthermore, if formatting is performed in the background, normal operation of the database system could continue unabated. Specification, page 7, lines 11-12. As a result, performance of the database system can be improved. Moreover, because formatting for the database system is done as needed-when the previous increment reaches a particular threshold-additional space remains available for other applications.

Hashimoto fails to teach or suggest a method, computer-readable medium, or system that asynchronously format an increment for a database system in response to occupation of a previous increment reaching a threshold of less than one hundred percent. Hashimoto discloses a magneto-optical disk formatting process that may be interrupted in order to process various requests. Hashimoto, col. 4, lines 30-31 and 50-51. The formatting process in Hashimoto is restarted "after the recording of the user data packet is completed" and after is determined whether formatting has been completed. Hashimoto, col. 8, lines 48-50 and FIG. 5. Stated differently, completion of writing in conjunction with a determination of whether formatting is

complete causes formatting to be restarted. Furthermore, the formatting process of Hashimoto apparently continues until the entire disk is formatted.

Applicant has found no mention in the cited portions of Hashimoto of formatting a particular increment *in response* to occupation of a previous increment reaching a threshold of *less than* one hundred percent. In contrast, Hashimoto apparently commences formatting of another increment either because the previous increment has been completed formatted (when formatting of Hashimoto is uninterrupted) or because a data packet is completely written (when formatting is interrupted to write the data packet). Hashimoto is thus devoid of mention of formatting an increment based on the occupation of a previous increment reaching a particular threshold. For example, if Hashimoto starts formatting another increment because the previous increment has completed formatting, then formatting does not depend upon the occupation of the previous increment. In the other case, Hashimoto starts formatting because of the completion of a writing process. There is no indication in Hashimoto that when writing the packet, a particular occupation threshold in the increment to which the data is written must be reached in order to start formatting the next increment. Instead, Hashimoto bases recommencing formatting on timing—when the data packet is completely written. Consequently, Hashimoto fails to teach or suggest a method, computer-readable medium, or system that asynchronously formats an increment in response to occupation of a previous increment reaching a threshold of less than one hundred percent. Hashimoto, therefore, fails to teach or suggest the method, computer-readable medium and system recited in claims 1, 9, and 17.

Moreover, Applicant notes that Hashimoto in view of Seamons also fails to teach or suggest the method, system, and computer-readable medium recited in claims 1, 9, and 17.

Seamons does not remedy the defects of Hashimoto. In particular, Seamons also fails to teach or suggest asynchronously formatting an increment based upon occupation of the previous increment reaching a threshold of less than one hundred percent. Seamons describes spreading the formatting process for a magneto-optical disk over time. Seamons, col. 5, lines 20-21. Seamons commences formatting a "predetermined time after the computer is turned on until there is a disk request from an application program." Seamons, col. 5, lines 30-32. Seamons thus interrupts formatting to service disk requests. The formatting is continued when the computer has been idle for a particular amount of time. Seamons, col. 5, lines 36-44. However, if the application is sufficient "disk-intensive that it uses up **all** of the formatted areas, . . . the system immediately formats additional space, rather than wait for disk-idle time." Seamons, col. 5, lines 47-52 (emphasis added). Seamons thus synchronously formats additional space when all formatted areas are used up.

Thus, Applicant can find no mention of asynchronously formatting a subsequent increment based upon the previous increment reaching a particular threshold of less then one hundred percent. Instead, Seamons bases the formatting either on another event—an amount of idle time for the computer—or the previous increment being full. In other words, Seamons formats an area when occupation a previous area is one hundred percent. Thus, Seamons also fails to teach or suggest formatting an increment in response to a threshold of less than one hundred percent being reached for the previous increment. Thus, Seamons cannot remedy the defects of Hashimoto.

Because both the cited portions of Hashimoto and Seamons fail to teach or suggest asynchronously formatting a subsequent increment based upon the previous increment reaching a particular threshold of less then one hundred percent, any combination of the cited portions of Hashimoto and Seamons also fails to teach or suggest this feature. If the teachings of Seamons are

added to those of Hashimoto, the combination might allow the system of Hashimoto to commence formatting at a particular time after the computer is turned on. The combination may also temporarily stop formatting when a disk request is made. The combination might recommence formatting either after a packet has been written, after the computer has been idle for a particular time, or if the formatted spaced is completely used. However, the combination would still fail to asynchronously format a subsequent increment *in response to* the occupation of a previous increment reaching a threshold of less than one hundred percent. Moreover, the combination would still fail to teach or suggest formatting that is specific to a database system. Consequently, Hashimoto in view of Seamons still fails to teach or suggest the method, computer-readable medium, and system recited in claims 1, 9, and 17. Accordingly, Applicant respectfully submits that claims 1, 9, and 17 are allowable over the cited references.

Claims 2, 10, and 18 depend upon independent claims 1, 9, and 17, respectively. Consequently, the arguments herein apply with full force to claims 2, 10, and 18. Accordingly, Applicant respectfully submits that claims 2, 10, and 18 are allowable over the cited references.

Independent claims 5, 13, and 21 recite a method, computer-readable medium, and system that begin asynchronously formatting at least one subsequent increment for the database file as a concurrent task of the database system as a background process when data are first stored in a previous increment. Thus, when data are first stored in an increment, formatting a subsequent is commenced. Thus, the method, computer-readable medium, and system recited in claims 5, 13, and 21 also do not format an increment based upon the end of a previous increment being reached. Instead, the method, computer-readable medium, and system of claims 5, 13, and 21 the *beginning* of a previous increment being reached being reached. As a result, access delays can be reduced or eliminated, and if formatting is performed in the background, normal operation of the

database system could continue unabated. Performance of the database system can thus be improved. Moreover, because formatting is only done as needed-when the previous increment is first written to, additional space is left for other applications.

Hashimoto and Seamons, separately or in combination, fail to teach or suggest asynchronous formatting of an increment for a database file when data are first stored in a previous increment. As discussed above, the formatting of Hashimoto and Seamons teach that formatting commences either when writing of a packet is complete (Hashimoto), when the computer has been idle for a particular time (Seamons) or when all of the formatted space has been used (Seamons). However, Applicant has found no discussion in the cited portions of either Hashimoto or Seamons of commencing any task related to a subsequently formatted increment when data are first stored in an increment. More specifically, the cited portions of neither Seamons nor Hashimoto describe asynchronously formatting an increment when data are first written to a previous increment. Consequently, any combination of Hashimoto and Seamons would fail to teach or suggest this feature. Hashimoto and Seamons, separately or in combination, thus fail to teach or suggest asynchronously formatting an increment for a database file when data are first written to a previous increment. Moreover, Hashimoto and Seamons apparently continue formatting until the entire disk is formatted. Thus, Hashimoto, Seamons, or their combination would not leave additional space for other applications. Consequently, Hashimoto and Seamons, separately or in combination, fail to teach or suggest the method, computer-readable medium, and system recited in claims 5, 13, and 21, respectively.

Independent claims 6, 14, and 22 recite a method, computer-readable medium, and system that format a first increment for a database system when a database file is opened and that asynchronously formats the at least one subsequent increment by continuously formatting the at

least one subsequent increment for the database as a background process. The method, computer-readable medium and system recited in claims 6, 14, and 22 can thus reduce or eliminate access delays and allow normal operation of the database system to continue. Performance of the database system can thus be improved.

Independent claims 6, 14, and 22 recite a method, computer-readable medium, and system that format a first increment for a database system when a database file is opened and asynchronously and continuously format subsequent increment(s) for the database file. Hashimoto and Seamons, separately or in combination, fail to teach or suggest these features. Seamons specifically times the start of formatting upon the computer being turned on. Applicant has found no mention in the cited portions of Hashimoto or Seamons of formatting a first increment when a database file is opened. Thus, any combination of Hashimoto and Seamons fail to teach or suggest commencing formatting when a database file is first opened. Furthermore, the cited portions of Hashimoto and Seamons are also devoid of mention of formatting subsequent increments continuously as a background process. Any combination of Hashimoto and Seamons, therefore, fails to teach or suggest formatting a first increment when a database file is opened in combination with formatting subsequent increments continuously. Consequently, Hashimoto in view of Seamons fails to teach or suggest the method, computer-readable medium, and system recited in claims 6, 14, and 22, respectively.

The Examiner also rejected claims 3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23, and 24 under 35 U.S.C. § 103 as being obvious in light of Hashimoto in view of Seamons.

Claims 4, 12, and 20 depend upon independent claims 1, 9, and 17, respectively. Consequently, the arguments herein apply with full force to claims 1, 9, and 17. Accordingly, Applicant respectfully submits that claims 4, 12, and 20 are allowable over the cited references.

Claims 7, 15, and 24 recite a method, computer-readable medium and system, respectively, which format a first increment for a database file, trigger the database system to asynchronously format subsequent increment(s) in response to occupation of the a previous increment reaching a threshold, and asynchronously format the subsequent increment(s) for the database file as a concurrent task. Thus, claims 7, 15, and 23 are analogous claims 1, 9, and 17. Consequently, the arguments herein apply with full force to claims 7, 15, and 23. Accordingly, Applicant respectfully submits that claims 7, 15, and 23 are allowable over the cited references.

Claims 8, 16, and 24 depend upon claims 7, 15, and 23, respectively. Consequently, the arguments herein apply with full force to claims 8, 16, and 24. Accordingly, Applicant respectfully submits that claims 8, 16, and 24 are allowable over the cited references.

Claims 31, 32, 33, 34, 35, 36, 37, 38, and 39 depend upon independent claims 1, 5, 7, 9, 13, 15, 17, 21, and 23, respectively. Consequently, the arguments herein apply with full force to claims 31-39. Accordingly, Applicant respectfully submits that claims 31-39 are allowable as presented.

Applicant's attorney believes that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicant's attorney at the telephone number indicated below.

Respectfully submitted,

SAWYER LAW GROUP LLP

January 30, 2004

Joseph A. Sawyer, Jr.
Attorney for Applicant(s)
Reg. No. 30,801
(650) 493-4540